CSCI E-101
Discussion 8

•••

Midterm Review

Midterm Review

1) Statistics Formulas covered in class
    a. (mean, std dev, covariance, Pearson R)

2) Python Basics
    a. Data types
    b. Expressions
    c. Relational and logical operators
    d. If-elif-else, for loop, while loop
    e. String slicing, string functions
    f. Call by values and reference

3) Python Modules and Functions covered in class
    a. Pandas, numpy, user-defined-functions,
    b. Pandas data frame, save to CSV, read from CSV
    c. MySQL connector, cursor and Pandas MySQL integration
    d. Executing SQL commands from Python
    i. SHOW, CREATE, DROP, SELECT, UNION, INSERT INTO, COMMIT
    e. Python filtering, replace, regex, pivot, pivot table
    f. All Pandas dataframe creation, data cleaning, merge concepts and techniques covered in class

Midterm Logistics

- Treat as a pencil and paper exam, try not to debug code in IDE unless you have extra time
- 90 minute exam, be mindful of the Canvas countdown timer, exam must be submitted at the end of the exam
- Read all questions prior to starting on the midterm. Midterm is designed to be lengthy to test the knowledge of content covered in class up to Module 7 (Data Visualization and Storytelling will not be on the midterm)
- Zoom proctored exam
  - Must have webcam on
  - You may use resources available to you without physically leaving the testing area
  - Some logistical questions may be answered via zoom chat
- Times offered for the exam https://www.surveymonkey.com/r/midtermsignup
  - Tuesday, November 2, from 8:00 am to 9:30 am ET (2nd choice)
  - Tuesday, November 2, from 3:00 pm to 4:30 pm ET (1st choice during class time)
  - Tuesday, November 2, from 5:30 pm to 7:00 pm ET (3rd choice)
  - Tuesday, November 2, from 11:00 pm to 12:30 am ET (Last choice)
- Mock Midterm
  - Midterm Practice Exam - "dress rehearsal" on Saturday, October 30th, at 10:00 pm ET

Midterm Formula Sheet

Statistical formulas covered in class

https://docs.google.com/document/d/1WJvmFPc9KQI1NtBYjWdjnmMukPQ-kTpcC9Cp__ie1Aw/edit?usp=sharing

Recommended Review

- Python basics
  - data types, expressions
  - operators; control structures; slicing (arrays string)
  - User defined functions
  - filtering, replace, regex, pivot table
  - data frames/series
- Stats
  - Review your formulas
  - mean, stddev, variance, covariance, z-score
- SQL
  - mysqlconnector, SHOW, CREATE, DROP, SELECT, UNION, INSERT INTO, COMMIT

Data Types

Type	*411*
str	sequence of characters
int, float	numbers
list	array++
tuple	immutable list
set	O(1) unique item list
dict	O(1) super fast key / value pair-based data structure

Python Basics

list comprehension: `[i for i in range(1,1000)]`

list comprehension with if/else: `[i/10 if i > 50 else 0 for i in range(0, 1000)]`

dict comprehension: `{i:i*2 for i in range(1,1000)}`

Slicing:

```
>>> a = [1,2,3,4,5]

>>> a[:1] + a[1:]
[1] +  [2, 3, 4, 5]

>>> a[-1]
5
```

List Comprehension

Original

```
my_list = []
For item in original_list:
   my_list.append(item * 2)
```

list comprehension:

```
my_list = [item * 2 for item in original_list]
```

Expressions

/ = float division ;

```
50/10 ---> (5.0)
```

// = floor (int) division

```
50//10 ---> (5)
```

** exponentiation

```
>>> 10**2
100
```

is

```
>>> a is None
False
```

not

```
>>> a is not None
True
```

in

```
>>> a = ["uno", "dos", "tres", "cuatro"]
>>> "one" in a
False
```

Built in Functions

abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

Operators

Operator	Purpose	Usage
&	returns 1 if both the corresponding bits are 1	a & b
|	returns 1 if any of the corresponding bits is 1	a | b
~	inverts all of the bits	~a
^	returns 1 if any of the corresponding bits is 1, but not both	a ^ b
>>	shifts the bits of ‘a’ to the right by ‘b’ no. of times	a >> b
<<	shifts the bits of ‘a’ to the left by ‘b’ no. of times	a << b

Pandas I/O

From DB

```
db_connection = sql.connect(host='127.0.0.1', database='employees', user=MYUSER, password=MYPASS)
salaries_df = pd.read_sql("""select * from salaries AS s inner join employees as e using (emp_no)
limit 1000""", con=db_connection)
```

To CSV

```
salaries_df.to_csv("test.csv",index=False)
```

From CSV

```
salaries2_df = pd.read_csv("test.csv")
```

Pandas Filtering

```
my_filter = (salaries_df["from_date"]>datetime.date(2000,1,1)) &
(salaries_df.first_name.str.contains("^G"))
salaries_df[my_filter]
```

	emp_no	salary	from_date	to_date	birth_date	first_name	last_name	gender	hire_date
14	10001	85112	2000-06-22	2001-06-22	1953-09-02	Georgi	Facello	M	1986-06-26
15	10001	85097	2001-06-22	2002-06-22	1953-09-02	Georgi	Facello	M	1986-06-26
16	10001	88958	2002-06-22	9999-01-01	1953-09-02	Georgi	Facello	M	1986-06-26
621	10063	71028	2000-04-05	2001-04-05	1952-08-06	Gino	Leonhardt	F	1989-04-08
622	10063	73393	2001-04-05	2002-04-04	1952-08-06	Gino	Leonhardt	F	1989-04-08
623	10063	74841	2002-04-04	9999-01-01	1952-08-06	Gino	Leonhardt	F	1989-04-08
758	10075	67492	2000-05-14	2001-01-15	1960-03-09	Gao	Dolinsky	F	1987-03-19

Pandas RegEx

```
(salaries_df[salaries_df['first_name']
             .str.contains(r'[aeiou]{2}')]
             .drop_duplicates('first_name')
             .head())
```

	emp_no	salary	from_date	to_date	birth_date	first_name	last_name	gender	hire_date
0	10001	60117	1986-06-26	1987-06-26	1953-09-02	Georgi	Facello	M	1986-06-26
30	10004	40054	1986-12-01	1987-12-01	1954-05-01	Chirstian	Koblick	M	1986-12-01
46	10005	78228	1989-09-12	1990-09-12	1955-01-21	Kyoichi	Maliniak	M	1989-09-12
106	10010	72488	1996-11-24	1997-11-24	1963-06-01	Duangkaew	Piveteau	F	1989-08-24
119	10012	40000	1992-12-18	1993-12-18	1960-10-04	Patricio	Bridgland	M	1992-12-18

```
(salaries_df['first_name']
             .str.extract(r'([aeiou]{2})')
             .drop_duplicates().head())
```

	0
0	eo
17	NaN
30	ia
46	oi
106	ua

Pandas Pivot

```
salaries_df['from_year'] = salaries_df['from_date'].dt.year
pd.pivot_table(salaries_df, values='salary', index=['emp_no', 'first_name'],
                    columns=['from_year']).fillna('')
```

	from_year	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998
emp_no	first_name														
10001	Georgi		60117.0	62102.0	66074.0	66596.0	66961.0	71046.0	74333.0	75286.0	75994.0	76884.0	80013.0	81025.0	81097.0
10002	Bezalel												65828.0	65909.0	67534.0
10003	Parto											40006.0	43616.0	43466.0	43636.0
10004	Chirstian		40054.0	42283.0	42542.0	46065.0	48271.0	50594.0	52119.0	54693.0	58326.0	60770.0	62566.0	64340.0	67096.0
10005	Kyoichi					78228.0	82621.0	83735.0	85572.0	85076.0	86050.0	88448.0	88063.0	89724.0	90392.0

Midterm Practice Problems

Python Basics Review

https://github.com/jcrogel/CSCI-S-101/tree/master/mid_term_review